

# **Megawin**

# **8051 ISP-ICP Programmer**

**&**

# **8051 ISP Writer U2**

## **User Manual**

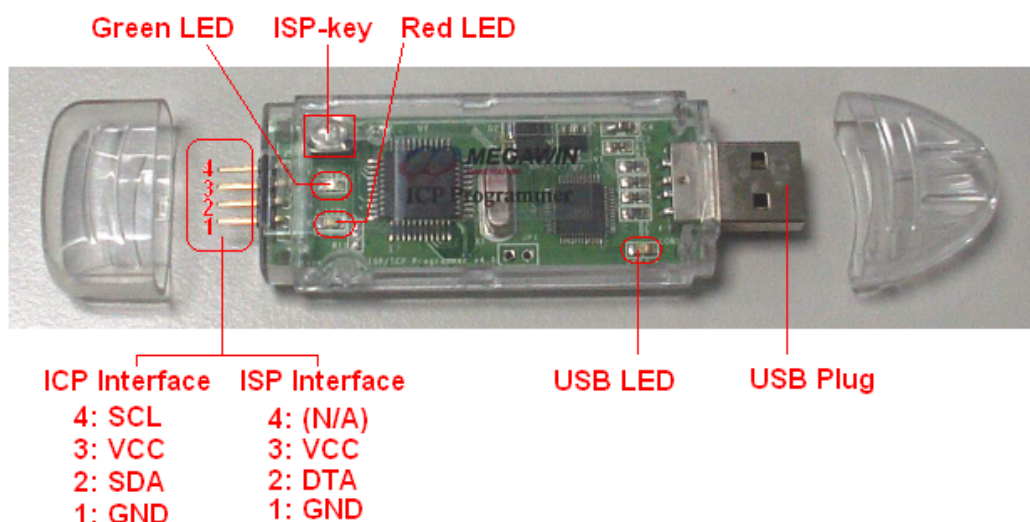
## Contents

|       |                                                        |    |
|-------|--------------------------------------------------------|----|
| 1     | Introduction .....                                     | 3  |
| 1.1   | Two-in-One Functions .....                             | 4  |
| 1.2   | Comparison between ISP and ICP .....                   | 6  |
| 2     | Chip Configuration for ISP .....                       | 7  |
| 3     | Install the ISP-ICP Programmer .....                   | 8  |
| 3-1   | Install the Driver .....                               | 8  |
| 3-2   | Install the AP .....                                   | 8  |
| 3-3   | Introduction to GUI of the PC-site AP .....            | 9  |
| 3-3-1 | ISP Programmer GUI for MPC89-series .....              | 11 |
| 3-3-2 | ISP Programmer GUI for MPC82/MG84/MG87-series .....    | 12 |
| 3-3-3 | ICP Programmer GUI for MPC82G516 .....                 | 13 |
| 3-3-4 | ICP Programmer GUI for MG84FL516 .....                 | 15 |
| 4     | Use the ISP-ICP Programmer .....                       | 17 |
| 4-1   | Operation Modes .....                                  | 17 |
| 4.1.1 | Mode-1: Connected between host and target system ..... | 17 |
| 4.1.2 | Mode-2: Connected to host only .....                   | 18 |
| 4.1.3 | Mode-3: Connected to target system only .....          | 18 |
| 4-2   | Act as an ISP Programmer .....                         | 19 |
| 4.2.1 | Download Programming Data to the ISP Programmer .....  | 19 |
| 4.2.2 | Update the Target .....                                | 19 |
| 4.2.3 | Dump the Contents in the Information Zone .....        | 19 |
| 4-3   | Act as an ICP Programmer .....                         | 20 |
| 4.3.1 | Download Programming Data to the ICP Programmer .....  | 20 |
| 4.3.2 | Update the Target .....                                | 20 |
| 4-4   | The Megawin Project File (MPJ File) .....              | 21 |
| 4.4.1 | Save to an MPJ File .....                              | 21 |
| 4.4.2 | Load an MPJ File .....                                 | 22 |
| 5     | Information Zone .....                                 | 23 |
| 5.1   | Definition of the Information Zone .....               | 23 |
| 5.2   | Dump the Information Data .....                        | 24 |
| 6     | Special Notes for ISP .....                            | 25 |
| 7     | Special Notes for ICP .....                            | 26 |
| 8     | About Information .....                                | 27 |
|       | Revision History .....                                 | 28 |

## **1 Introduction**

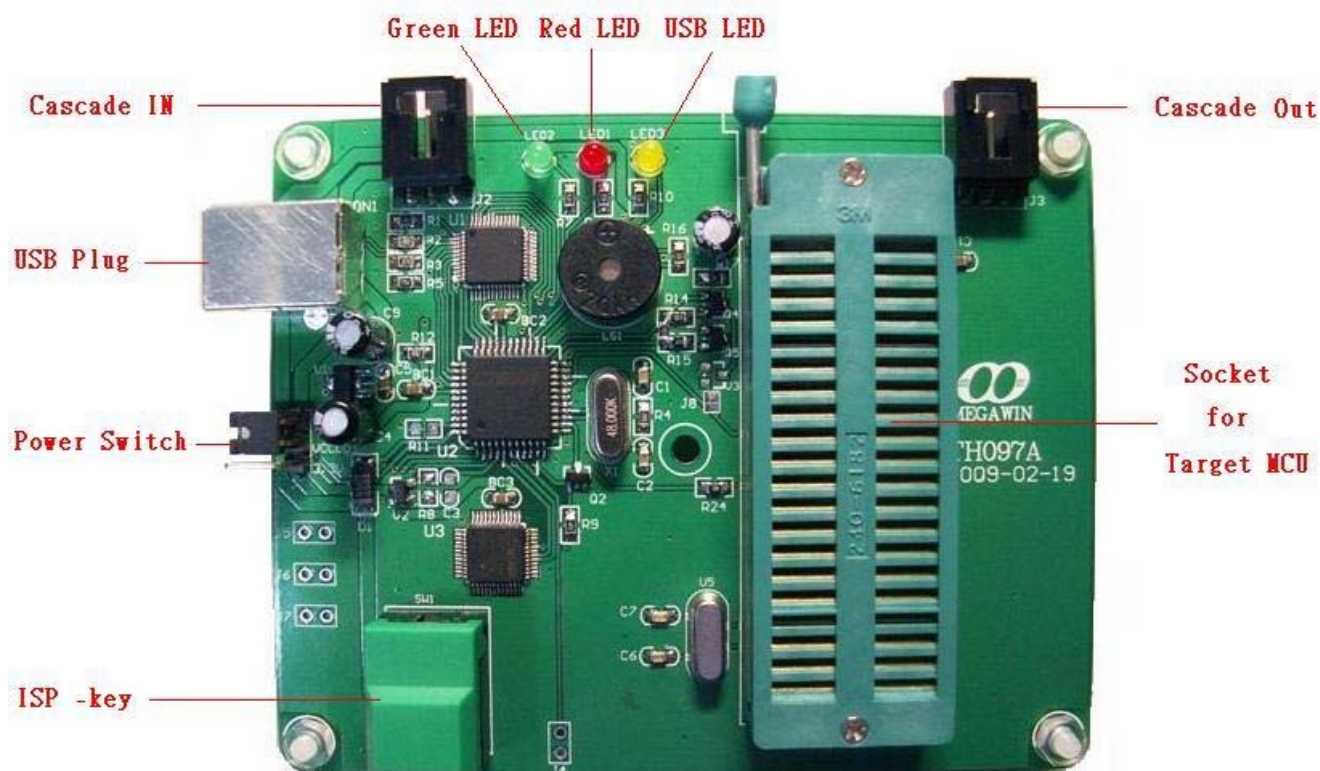
**ISP** is the acronym of **In-System Programming**, and **ICP** is the acronym of **In-Circuit Programming**. Both these two programming methods make it possible that the user can update the application code under the software control without removing the mounted MCU chip from the actual end product. The tool “Megawin 8051 ISP-ICP Programmer”, see the following picture, integrates these two functions into a USB stick. That is it can function as an ISP Programmer and an ICP Programmer. In addition, because the programming data to be programmed to the target can be saved in the programmer’s non-volatile storage, this programmer is able to work stand-alone without host (PC) intervention. This feature is especially useful in the field without a PC.

### **Picture of the ISP-ICP Programmer**



The tool of “Megawin 8051 ISP Writer U2”, see the following picture, can be treated as an ISP Programmer with Target System. It is used for mass production (Cascade to Gang4 or Gang8 mode) to update the MCU chip application code before mount to the end product.

### Picture of the ISP Writer U2



## 1.1 Two-in-One Functions

### ISP Programmer

When acting as an ISP Programmer, it functions like a bridge between the host, which provides the new programming data, and the target MCU, which has the loader program running inside. The loader program is the so-called “ISP-code” and should be pre-programmed in the ISP-memory of the target. When powered on, the target boots from the ISP-memory and executes the ISP-code to check if the ISP operation is requested. If the ISP is requested, the target receives the programming data from the Programmer and programs into the AP-memory by in-system programming method. After ISP processing is completed, the target will re-boot from the AP-memory to run the new application code when the Programmer is plugged out of the target; If the ISP is not requested, the target will directly re-boot from the AP-memory for normal running of the application code.

### ICP Programmer

When acting as an ICP Programmer, it actually functions fully like a universal programmer except it adopts a serial interface with only four pins used for programming. So, there is no need to have a loader program embedded inside. It provides all the programming functions which a universal programmer can support, such as:

- (1) erase the device,
- (2) program the device,

- (3) verify the device, and
- (4) program all the device H/W options, including the security bits.

## 1.2 Comparison between ISP and ICP

The following table shows the comparison between ISP and ICP and the parts that support them.

| Compared Items                             | ISP                                                                                                                                                             | ICP                                                      |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| Parts that support ISP or ICP              | MPC89L(E)51/52/53<br>MPC89L(E)54/58/515<br>MPC82L(E)52<br>MPC82L(E)54<br>MPC82G516 <sup>Note1</sup><br>MG84FL54<br>MG84FL516 <sup>Note1</sup><br>MG87FL(E)51/52 | MPC82G516 <sup>Note1</sup><br>MG84FL516 <sup>Note1</sup> |
| Erase, Program and Verify                  | Yes                                                                                                                                                             | Yes                                                      |
| Update target's H/W option                 | Partial or None                                                                                                                                                 | Yes                                                      |
| Interface                                  | GND/DTA/VCC <sup>Note2</sup>                                                                                                                                    | GND/SDA/VCC/SCL <sup>Note2</sup>                         |
| Chip configuration before using ISP or ICP | ISP-code pre-programmed & HWBS enabled                                                                                                                          | Not Needed                                               |

*Note:*

- 1. So far, only the MPC82G516 and MG84FL516 support both ISP and ICP.*
- 2. ISP interface always uses P3.1 as DTA pin while ICP interface uses dedicated SDA and SCL pins.*

## **2 Chip Configuration for ISP**

To use the ISP function, the user should configure the target MCU by use the “Insert ISP-code” function in “Megawin 8051 Writer” or “Megawin 8051 Writer U1”. And set the **ISP-memory** with 1K bytes (or 1.5K bytes for MPC82L(E)54 ) and make **HWBS** or **HWBS2** option enabled.

### **Note:**

*To let users easily use the ISP function, the Megawin 8051 products will have the following factory setting:*

- (1) ISP-memory is configured with 1K (or 1.5K) bytes and “HWBS” option is enabled.*
- (2) The Megawin-provided standard ISP-code is pre-programmed.*

*So, the user has no need to do the chip configuration before using the ISP function.*

*\*\*\* Contact Megawin for detailed product information.*

### **3 Install the ISP-ICP Programmer**

#### **3-1 Install the Driver**

Plug the ISP-ICP Programmer into the PC's USB port, and do as follows when the monitor shows a prompt about new hardware found.

- 1) Select **No, not this time**, click **Next**.
- 2) Select **Install from a list or specific location**, click **Next**.
- 3) Select **Search for the best driver in these locations** and **Include this location in the search**, click **Browse**.
- 4) Locate the driver folder [(2) PC-site Driver], click **OK**.
- 5) Click **Next**. The driver installation starts.
- 6) Click **Finish** when the installation completes.

To check if the Programmer was correctly installed, follow the listed steps:

- 1) Open the **My Computer** folder.
- 2) Open the **Control Panel** folder.
- 3) Open the **System**.
- 4) Click on the **Hardware** tab at the top of the dialog box, then click on the **Device Manager**.
- 5) Click on the plus sign in front of the **Universal Serial Bus Controllers** to check the device listing.

If the installation was completed successfully, you may find an entry, **Megawin 8051 ISP-ICP Programmer**, in the listing.

#### **3-2 Install the AP**

Run "**Setup.exe**" (in the [(1) PC-site AP] folder) to install the application program for the ISP-ICP Programmer on your PC. Using its default installing setting, you will find the item "**Megawin Utilities \ Megawin 8051 ISP-ICP Programmer (v?.??)**" appearing in the Windows START-menu.

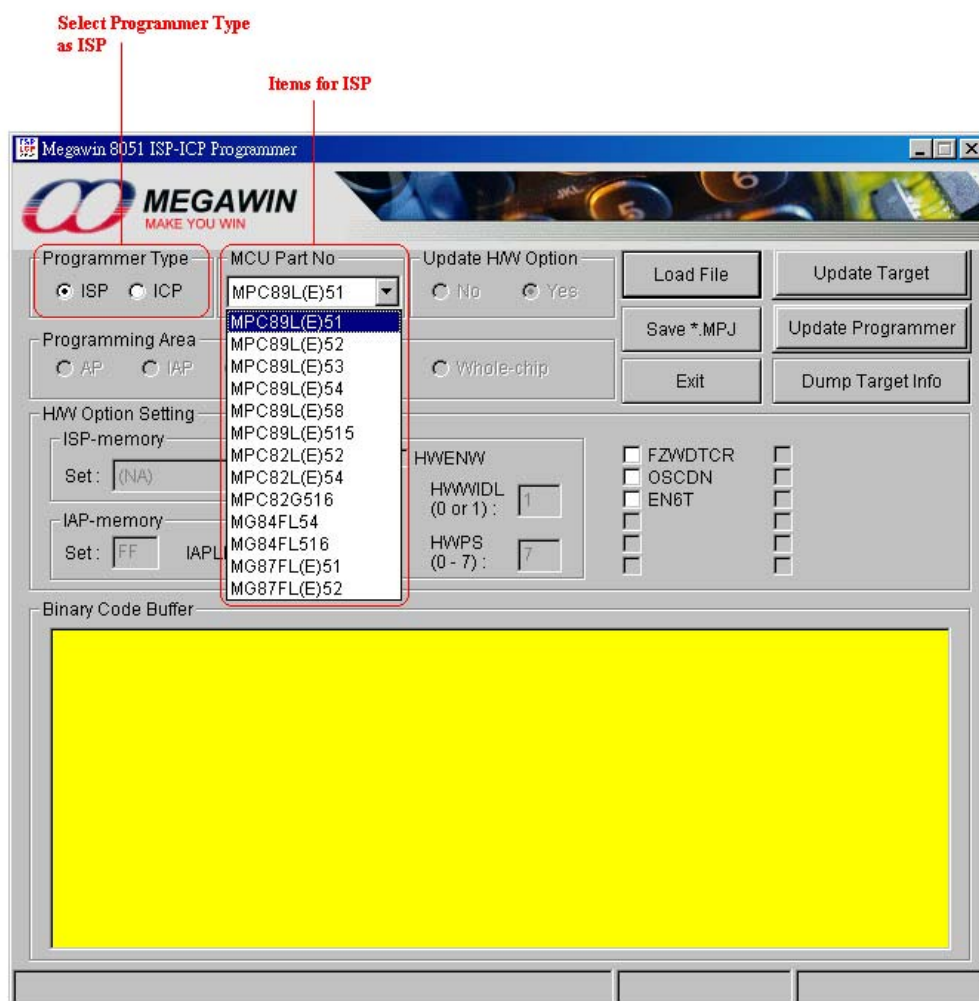
*(Note: the v?.?? means the current version and may be upgraded in the future.)*

### 3-3 Introduction to GUI of the PC-site AP

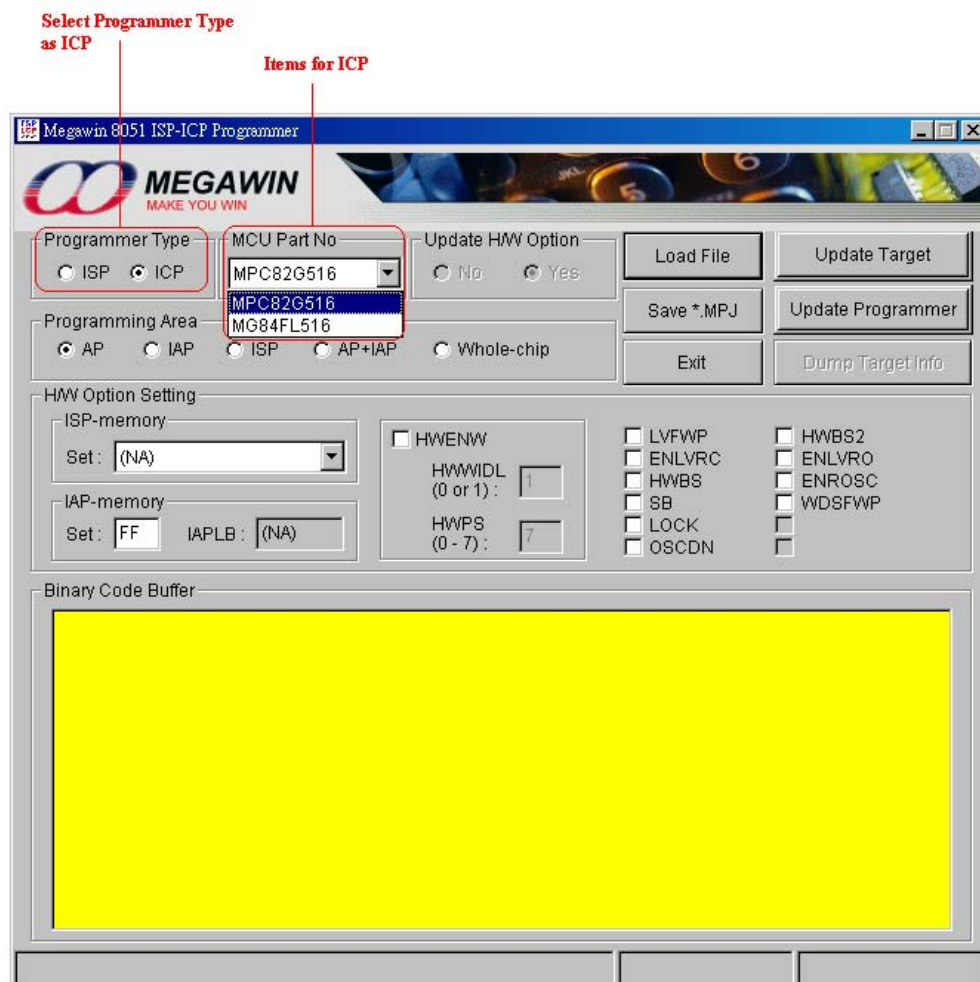
GUI means “Graphic User Interface” of the Application Program running in the Windows.

The PC-site software AP (Application Program) integrates both functions of the ISP Programmer and the ICP Programmer. The first thing the user needs to do is to select the “Programmer Type” when the AP is opened. See the following figures for these two programmer types.

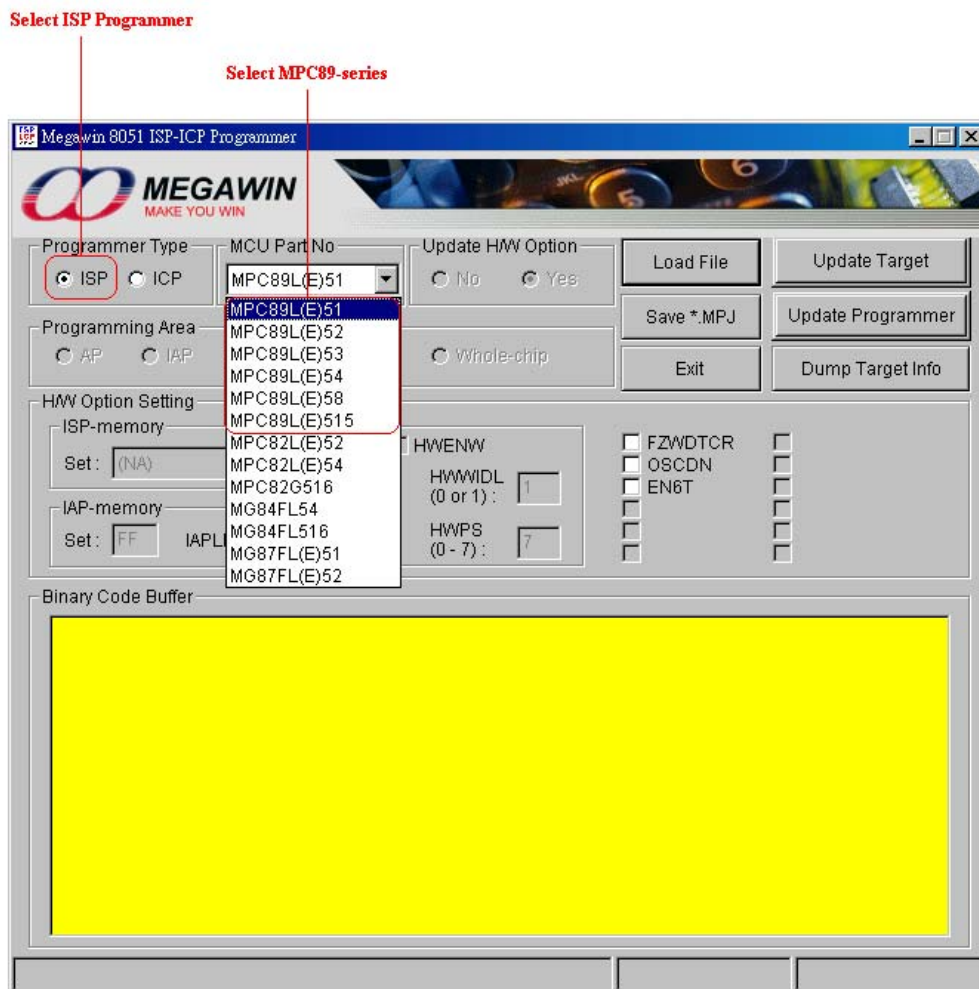
#### ISP Programmer Type



## ICP Programmer Type



### 3-3-1 ISP Programmer GUI for MPC89-series



#### About the H/W Option Setting

The user should always configure proper H/W Option before clicking “Update Target” or “Update Programmer”.

##### FZWDTCR:

[enabled]: The WDTCR register will be initialized to its reset value (0x00) *only by power-on reset.*  
*(For example, if WDTCR=0x2D, it still keeps at 0x2D rather than 0x00 after RST-pin, S/W or WDT reset.)*

[disabled]: The WDTCR register will be initialized to its reset value (0x00) by all reset (including power-on, RST-pin, S/W and WDT reset).

##### OSCDN:

[enabled]: If the XTAL frequency is less than 25MHz, this option can be enabled to reduce the internal oscillating gain for lower EMI.

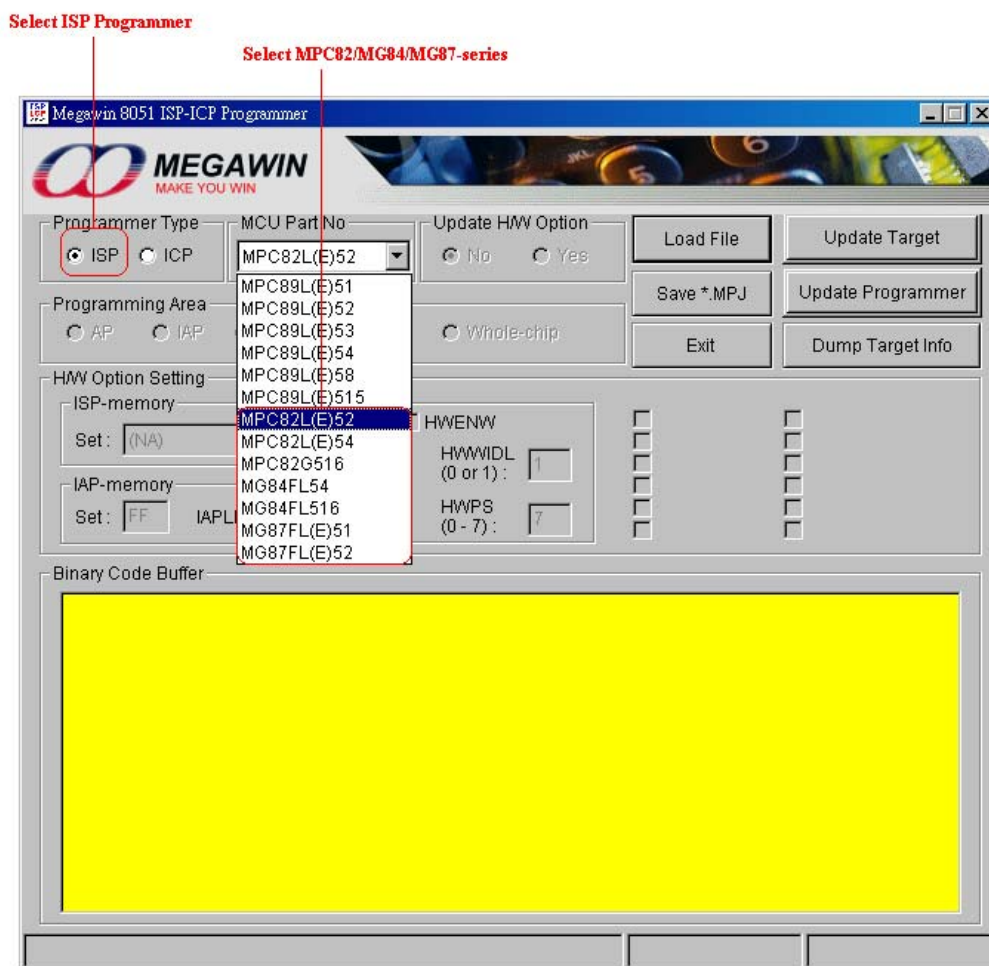
[disabled]: Normal oscillating gain.

##### EN6T:

[enabled]: MCU runs at 6T mode (6 clocks per machine-cycle, double speed compared to a traditional 8051)

[disabled]: MCU runs at 12T mode (12 clocks per machine-cycle, like a traditional 8051)

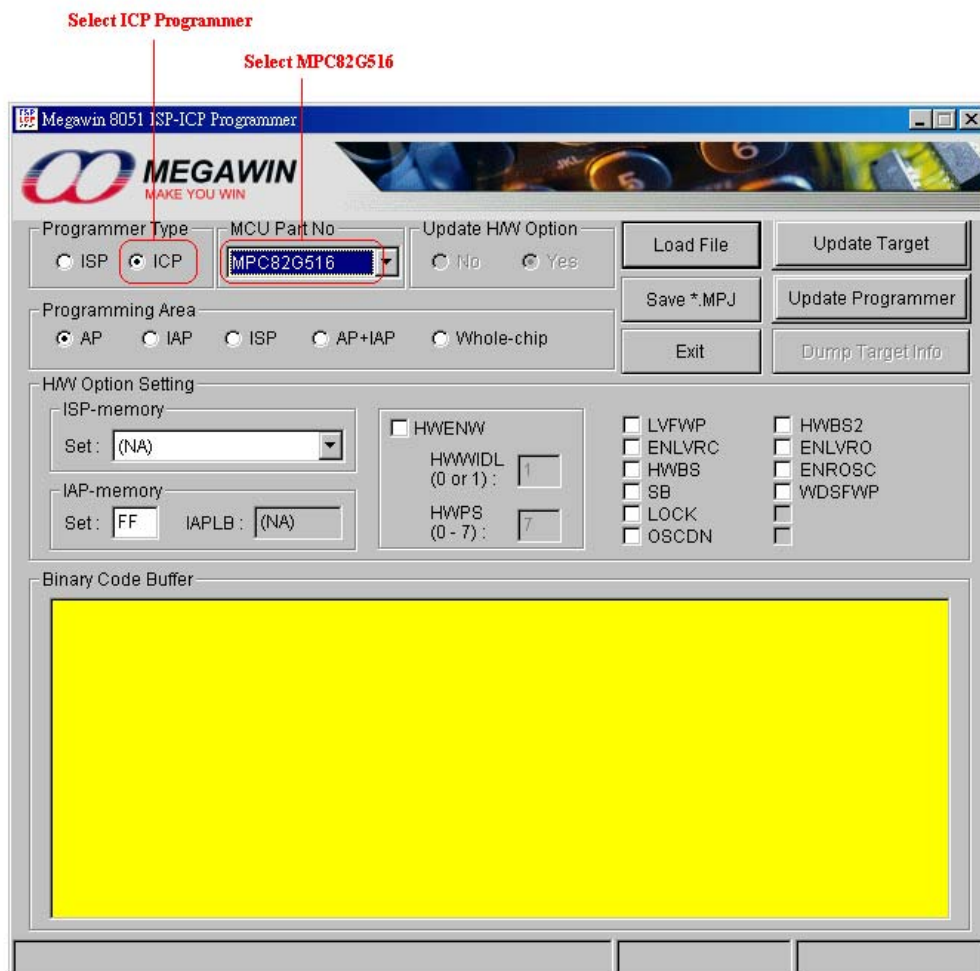
### 3-3-2 ISP Programmer GUI for MPC82/MG84/MG87-series



#### About the H/W Option Setting

For these series, the user can not update the H/W Option by ISP.

### 3-3-3 ICP Programmer GUI for MPC82G516



#### About the H/W Option Setting

The user should always configure proper H/W Option before clicking "Update Target" or "Update Programmer".

##### ISP-memory:

Select the size you want.

##### IAP-memory:

Select the lower boundary address you want. For example, if 0x5A00 is wanted, then just key in "5A", and the IAP-memory lower boundary address will automatically displayed in the IAPLB box.

##### LVFWP:

[enabled]: Enable LVFWP (Low-Voltage Flash Write Protection) while IAP or ISP programming.  
 [disabled]: Disable LVFWP.

##### ENLVRC:

[enabled]: Enable hardware to generate low voltage reset when V30-pin voltage drops below 2.4V.  
 [disabled]: Disable low voltage reset.

**HWBS:**

[enabled]: When power-on, MCU will boot from ISP-memory if ISP-memory is configured.  
 [disabled]: MCU always boots from AP-memory.

**SB:**

[enabled]: Code dumped on a universal Writer or Programmer is scrambled for security, but Device ID can be read normally. *It is strongly recommended that the LOCK should also be enabled when SB is enabled.*  
 [disabled]: Not scrambled.

**LOCK:**

[enabled]: Code dumped & Device ID read on a universal Writer or Programmer is locked to 0xFF for security.  
 [disabled]: Not locked.

**OSCDN:**

[enabled]: Oscillating gain is reduced down for EMI reduction.  
 [disabled]: Normal gain.

**HWBS2:**

[enabled]: Like HWBS, the reset from RST-pin can also cause MCU to boot from ISP-memory.  
 [disabled]: Where MCU boots from is determined by **HWBS**.

**ENLVRO:**

[enabled]: Enable MCU to generate low voltage reset when VDD-pin voltage drops below 3.7V.  
 [disabled]: No low voltage reset.

**ENROSC:**

[enabled]: Enable built-in RC oscillator.  
 [disabled]: Disable built-in RC oscillator.

**WDSFWP:**

[enabled]: The special function register WDTCR will be write-protected except the bit CLRW.  
 [disabled]: The special function register WDTCR is free to be written by software.

**HWENW (accompanied with arguments HWWIDL and HWPS[2:0]):**

[enabled]: Automatically enable Watch-dog Timer by the hardware when the MCU is powered up.

It means that:

In the WDTCR register, the hardware will automatically:

- (1) set **ENW** bit,
- (2) load **HWWIDL** into **WIDL** bit, and
- (3) load **HWPS[2:0]** into **PS[2:0]** bits.

For example:

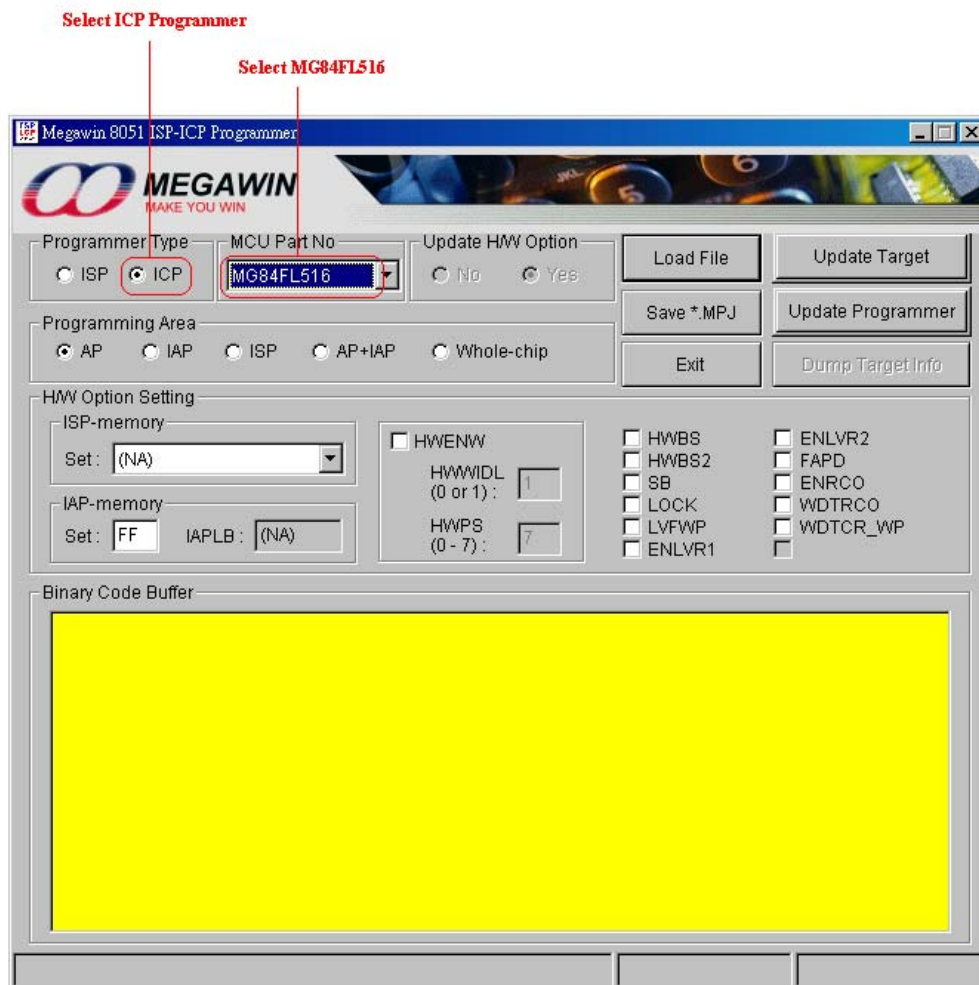
If **HWWIDL** and **HWPS[2:0]** are programmed to be 1 and 5, respectively, then **WDTCR** will be initialized to be 0x2D when MCU is powered up, as shown below.

**WDTCR** (Watch-Dog-Timer Control Register)

| 7   | 6 | 5          | 4     | 3                | 2   | 1                   | 0   |
|-----|---|------------|-------|------------------|-----|---------------------|-----|
| WRF | - | ENW        | CLRWW | WIDL             | PS2 | PS1                 | PS0 |
|     |   | ↑ set<br>1 |       | ↑ load<br>HWWIDL |     | ↑ load<br>HWPS[2:0] |     |

[disabled]: No action on Watch-dog Timer when the MCU is powered up.

### 3-3-4 ICP Programmer GUI for MG84FL516



#### About the H/W Option Setting

The user should always configure proper H/W Option before clicking “Update Target” or “Update Programmer”.

##### ISP-memory:

Select the size you want.

##### IAP-memory:

Select the lower boundary address you want. For example, if 0x5A00 is wanted, then just key in “5A”. The IAP-memory lower boundary address will automatically displayed in the IAPLB box.

##### HWBS:

[enabled]: When power-on, MCU will boot from ISP-memory if ISP-memory is configured.

[disabled]: MCU always boots from AP-memory.

##### HWBS2:

[enabled]: In addition to power-on-reset, the reset from RST-pin will also force MCU to boot from ISP-memory if ISP-memory is configured.

[disabled]: Where MCU boots from is determined by **HWBS**.

**SB:**

[enabled]: Code dumped on a universal Writer or Programmer is scrambled for security, but Device ID can be read normally. *It is strongly recommended that the LOCK should also be enabled when SB is enabled.*  
 [disabled]: Not scrambled.

**LOCK:**

[enabled]: Code dumped & Device ID read on a universal Writer or Programmer is locked to 0xFF for security.  
 [disabled]: Not locked.

**LVFWP:**

[enabled]: Enable LVFWP (Low-Voltage Flash-Write Protection) while IAP or ISP programming.  
 [disabled]: Disable LVFWP.

**ENLVR1:**

[enabled]: MCU to generate low voltage reset when VDD-pin voltage drops below LVD1.  
 [disabled]: No low voltage reset.

**ENLVR2:**

[enabled]: MCU to generate low voltage reset when VDD-pin voltage drops below LVD2.  
 [disabled]: No low voltage reset.

**FAPD:**

[enabled]: Enable Flash-Auto-Power-Down to save power while normal operating.  
 [disabled]: Disable Flash-Auto-Power-Down.

**ENROSC:**

[enabled]: Enable built-in RC oscillator (6MHz).  
 [disabled]: Disable built-in RC oscillator.

**WDTRCO:**

[enabled]: Enable the RC oscillator, and select RCosc as OSCin.  
 [disabled]: Disable the RC oscillator, and select XTALosc as OSCin.

**WDTCR\_WP:**

[enabled]:  
 If CPU runs in AP-memory, the register WDTCR will be software-write-protected except the bit CLRW.  
 If CPU runs in ISP-memory, the register WDTCR will be software-write-protected except the bits CLRW, PS2, PS1 and PS0.  
 [disabled]: The register WDTCR can be freely written by software.

**HWENW (accompanied with arguments HWWIDL and HWPS[2:0]):**

[enabled]: Automatically enable Watch-dog Timer by the hardware when the MCU is powered up.

It means that:

In the WDTCR register, the hardware will automatically:

- (1) set **ENW** bit,
- (2) load **HWWIDL** into **WIDL** bit, and
- (3) load **HWPS[2:0]** into **PS[2:0]** bits.

For example:

If **HWWIDL** and **HWPS[2:0]** are programmed to be 1 and 5, respectively, then **WDTCR** will be initialized to be 0x2D when MCU is powered up, as shown below.

**WDTCR** (Watch-Dog-Timer Control Register)

| 7   | 6 | 5          | 4    | 3                | 2          | 1                   | 0          |
|-----|---|------------|------|------------------|------------|---------------------|------------|
| WRF | - | <b>ENW</b> | CLRW | <b>WIDL</b>      | <b>PS2</b> | <b>PS1</b>          | <b>PS0</b> |
|     |   | ↑ set<br>1 |      | ↑ load<br>HWWIDL |            | ↑ load<br>HWPS[2:0] |            |

[disabled]: No action on Watch-dog Timer when the MCU is powered up.

## 4 Use the ISP-ICP Programmer

### 4-1 Operation Modes

There are three operation modes for the ISP-ICP Programmer based on its connection conditions.

#### 4.1.1 Mode-1: Connected between host and target system

In this condition, the ISP-ICP Programmer works with the PC-site AP being executed. Three main buttons can be clicked: (1) The **“Update Programmer”** button, which is used to download the programming data (including Part No., user’s application code and H/W option) into the non-volatile storage of the Programmer. (2) The **“Update Target”** button, which further programs the new application code and H/W option into the Target MCU in addition to those the “Update Programmer” button does. (3) The **“Dump Target Info”** button, which dumps the Target Information Data described in [Section 5](#).

The user should select correct **“Part No.”**, then click **“Load File”** to load the new application code, and configure the H/W options (if have) before clicking the “Update Programmer” or “Update Target” button. The ISP-key can also be used to start the ISP or ICP processing after the “Update Programmer” is completed. Of course, the user may load the *MPJ* file (described in [Section 4.4](#)) to restore all the programming data to the GUI of the AP.

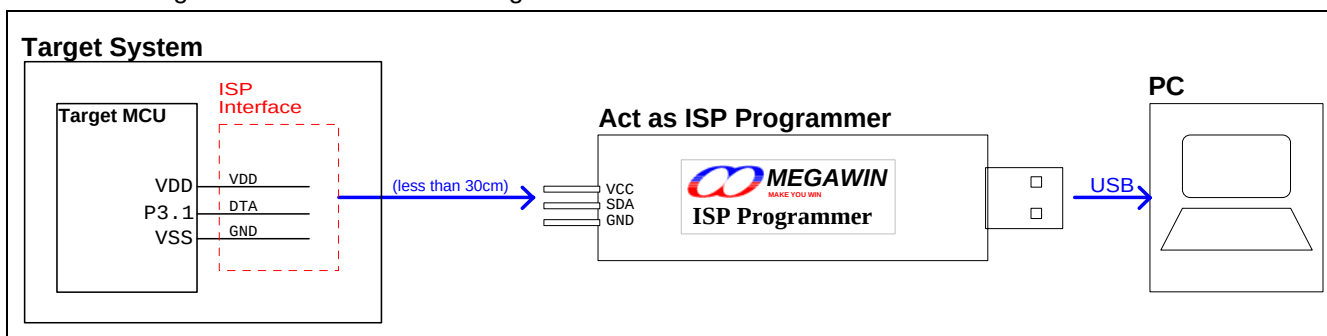
The following figures show the connection diagram:

(1) When the Programmer acts as an ISP Programmer, *don’t power on the target system before connecting ok*. After connecting ok, then the user’s system can be powered on. At this time, the target MCU keeps running in the ISP-memory for ISP processing. When ISP is finished, disconnect this Programmer from the target system to let the target MCU run the new application code.

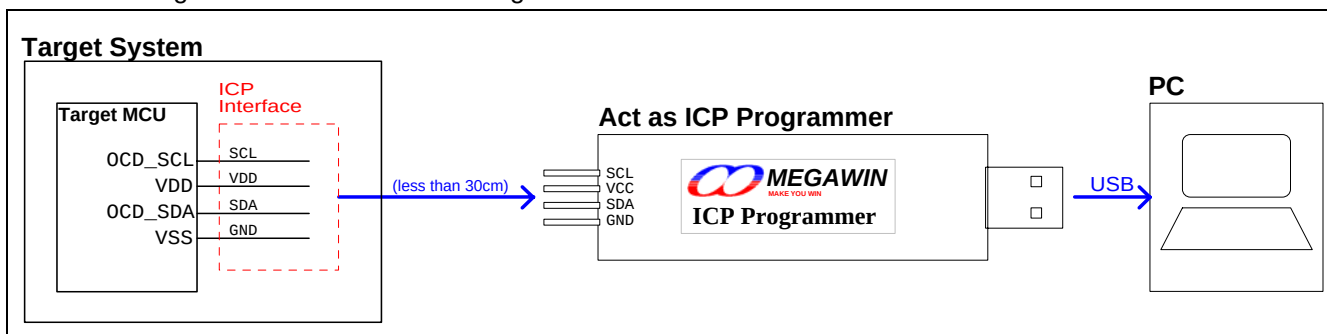
(2) When the Programmer acts as an ICP Programmer, the target system can be powered on *before* or *after* the connecting. And the target MCU will automatically run the new application code without need of disconnecting from the target system after ICP is finished.

The LEDs show the processing result. If ISP/ICP processing succeeds, the green LED will turn on, otherwise the red LED will turn on.

When the Programmer acts as an ISP Programmer:

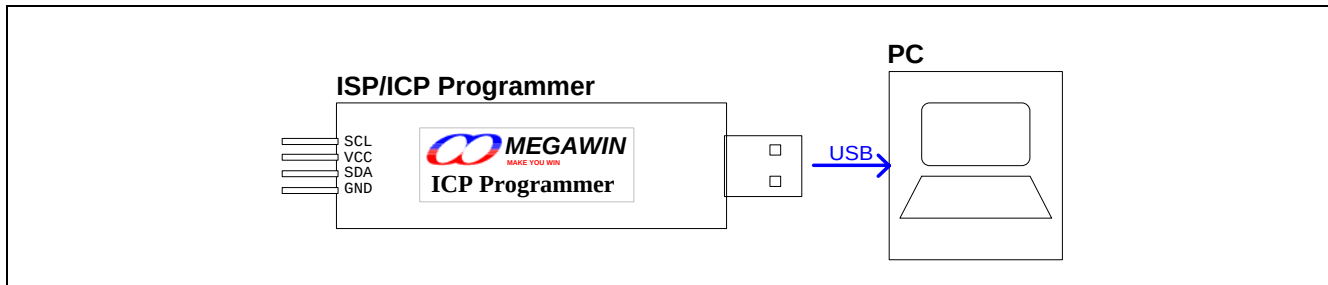


When the Programmer acts as an ICP Programmer:



#### 4.1.2 Mode-2: Connected to host only

In this condition, the ISP-ICP Programmer works with the PC-site AP being executed and only the “**Update Programmer**” button can be clicked. User can download the programming data into the non-volatile storage in the Programmer for later stand-alone operation.



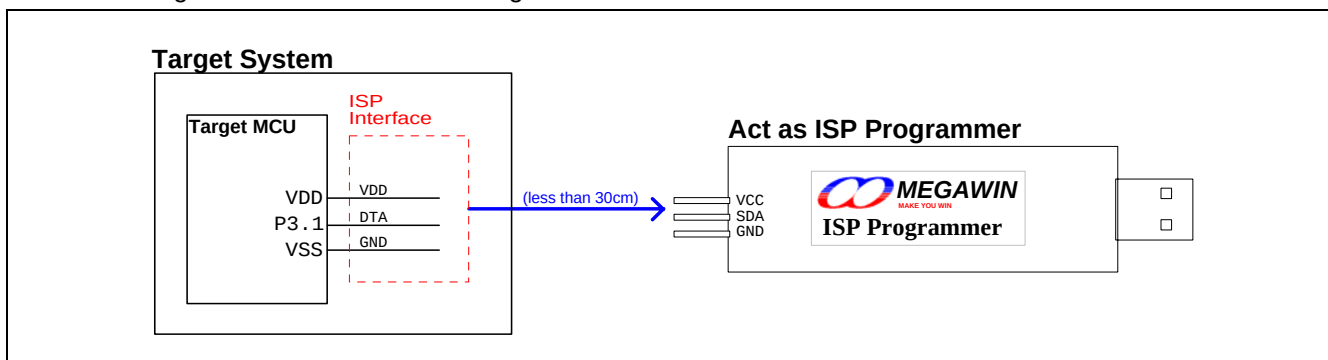
#### 4.1.3 Mode-3: Connected to target system only

In this condition, the Programmer works stand-alone without the AP’s intervention.

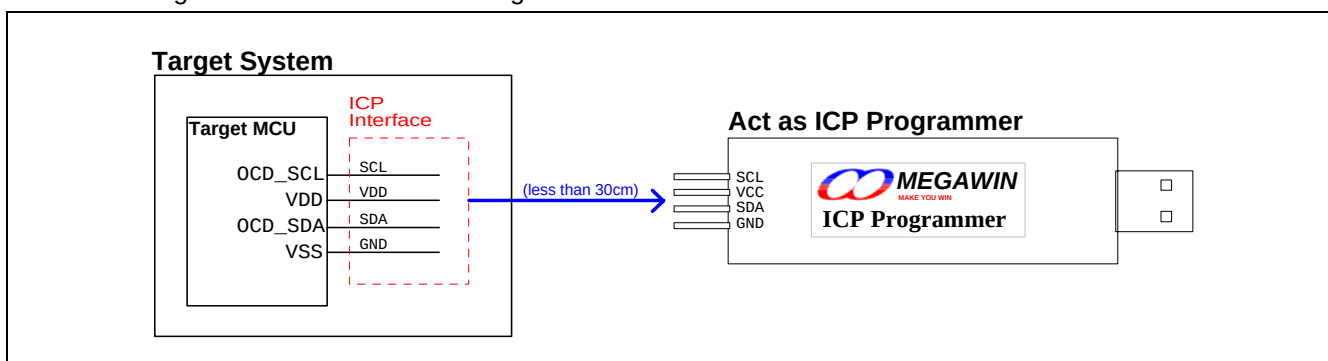
(1) When acting as an ISP Programmer, connect the Programmer to the target system before the system is powered up. Then, power on the system, and press the **ISP-key** to start ISP processing. The green and red LEDs show the processing result. Now, the user can disconnect the Programmer to let the system start running the new application code.

(2) When acting as an ICP Programmer, connect the Programmer to the target system at any time regardless of the system’s power state. Then press the **ISP-key** to start ICP processing while the system is powered on. The green and red LEDs show the processing result. And once the processing is finished, the system will automatically run the new application code.

When the Programmer acts as an ISP Programmer:



When the Programmer acts as an ICP Programmer:



## **4-2 Act as an ISP Programmer**

### **4.2.1 Download Programming Data to the ISP Programmer**

How to download new programming data (including application code & H/W options) into the Programmer?

Step 1: Select “Programmer Type” as *ISP Programmer*, and select the wanted “MCU Part No”.

Step 2: Click “Load File”, both HEX format and BIN format are acceptable, and the code size is based on its binary format.

For **MPC89L(E)51/52/53**, the maximum code size is 15K-1K=**14K** bytes, which includes IAP data.

For **MPC89L(E)54/58/515**, the maximum code size is 63K-1K=**62K** bytes, which includes IAP data.

For **MPC82L(E)52**, the maximum code size is 8K-1K=**7K** bytes, which includes IAP data.

For **MPC82L(E)54**, the maximum code size is 15.5K-1.5K=**14K** bytes, which includes IAP data.

For **MPC82G516**, the maximum code size is 64K-1K=**63K** bytes, which includes IAP data.

For **MG84FL54**, the maximum code size is 16K-1K=**15K** bytes, which includes IAP data.

For **MG84FL516**, the maximum code size is 64K-1K=**63K** bytes, which includes IAP data.

For **MG87L(E)51/52**, the maximum code size is 8K-1K=**7K** bytes, which includes IAP data.

Where, “minus 1K” (or 1.5K for MPC82L(E)54 ) means subtracting the space of ISP-memory.

Step 3: Configure the wanted H/W Option. (Only available for MPC89-series.)

Step 4: Click “Update Programmer”.

Step 5: Set the Maximum number of devices programmed, zero for no limits

*Note: In Step 2, you can load an MPJ file (refer to [Section 4.4](#)), then Step 3 is not necessary.*

### **4.2.2 Update the Target**

How to update the target?

Step 1~3: The same as previous Step1~3..

Step 4: Click “Update Target”.

### **4.2.3 Dump the Contents in the Information Zone**

How to dump the 256 bytes of target “Information Data”? (Please refer to [Section 5](#))

Step 1: Select “Programmer Type” as an *ISP Programmer*.

Step 2: Click “Dump Target Info”.

## **4-3 Act as an ICP Programmer**

### **4.3.1 Download Programming Data to the ICP Programmer**

How to download new programming data (including application code & H/W options) into the Programmer?

- Step 1: Select “Programmer Type” as *ICP Programmer*, and select the wanted “MCU Part No”.
- Step 2: Click “Load File”, both HEX format and BIN format are acceptable, and the code size is based on its binary format.
- Step 3: Configure the wanted H/W Option.
- Step 4: Click “Update Programmer”.
- Step 5: Set the Maximum number of devices programmed, zero for no limits

*Note: In Step 2, you can load an MPJ file (refer to [Section 4.4](#)), then Step 3 is not necessary.*

### **4.3.2 Update the Target**

How to update the target?

- Step 1~3: The same as previous Step1~3..
- Step 4: Click “Update Target”.

## 4-4 The Megawin Project File (MPJ File)

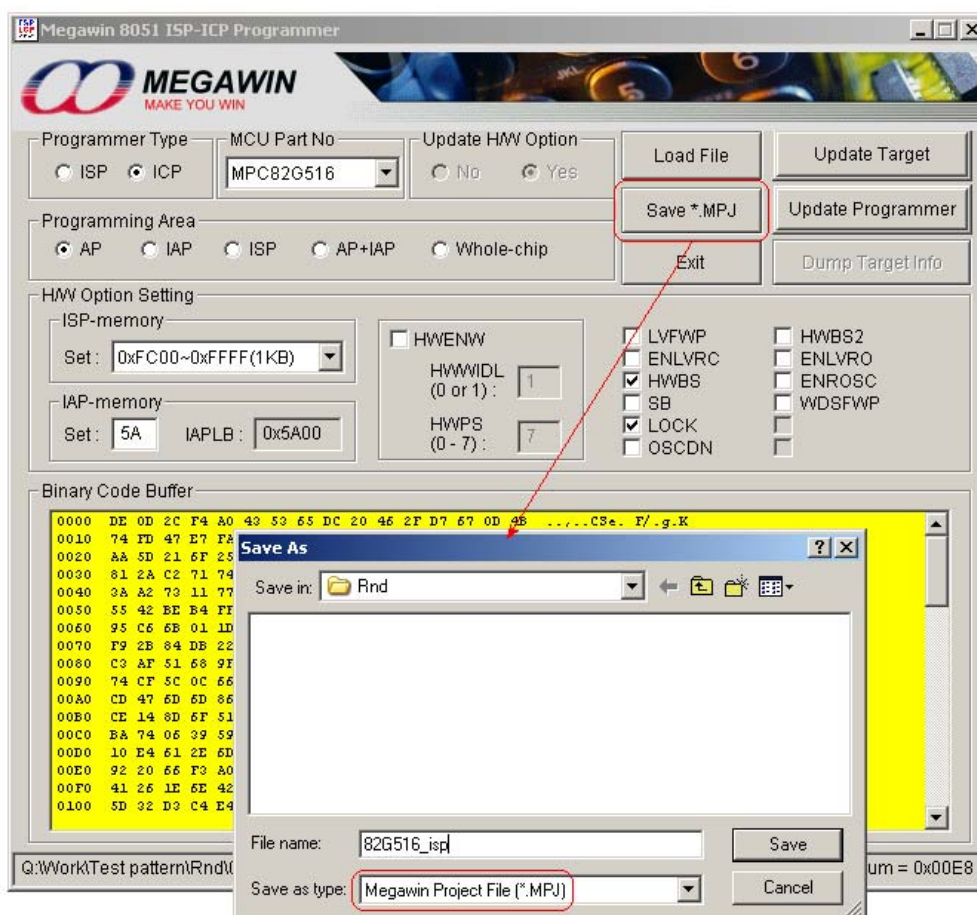
In addition to the application code, the Programmer may program the H/W options to the target MCU at each time the "Update Target" button (on the PC-site AP) is clicked or the ISP-key (on the Programmer) is pressed. And, maybe the operating of application code depends on the H/W options (for example, EN6T), so the one who takes the programming task must set the correct H/W options in addition to loading the correct application code. To prevent from any mistake (for example, forget to set the correct H/W options), all the relevant programming information should be saved together in one file, which we called the **MPJ** file.

### 4.4.1 Save to an MPJ File

The MPJ file includes all the relevant programming information appearing on the GUI, they are:

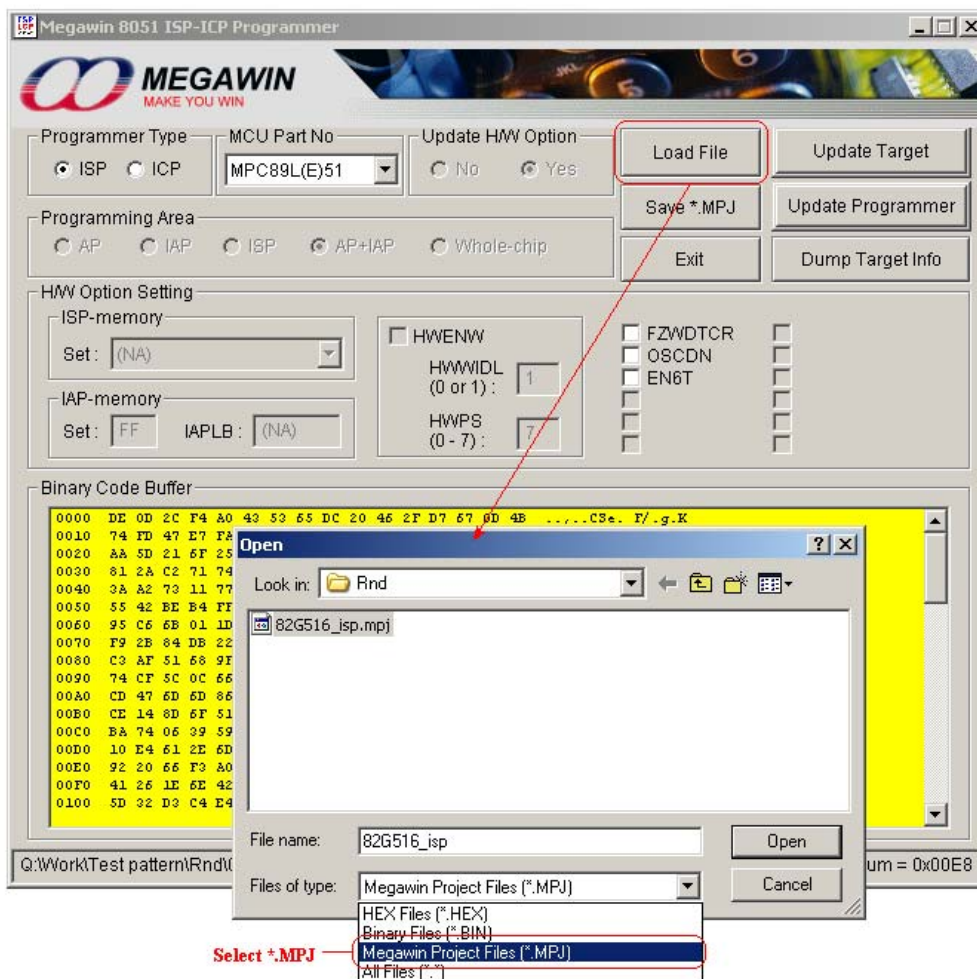
- (1) Programmer Type,
- (2) MCU Part No,
- (3) Control of Update H/W Option,
- (4) Programming Area,
- (5) H/W Option Setting, and
- (6) Binary Code Buffer.

Click the button "**Save \*.MPJ**" to save all the programming information in an MPJ file, as shown below.



#### 4.4.2 Load an MPJ File

Click "Load File" and select **Megawin Project Files (\*.MPJ)** to select the MPJ file you want. When the MPJ file is opened, all the relevant programming information will be retrieved and restored to the GUI of the AP. The following figure shows how to open an MPJ file.



## 5 Information Zone

When the H/W option *LOCK* or *SB* is enabled for code security, there is no way to read out the contents of the target by neither a universal programmer nor the ISP Programmer. For the purpose of user's product failure analysis in the future, a special Flash area called **Information Zone** is defined to be readable by only the ISP Programmer even the target is locked or scrambled. The user may put some useful product information there for future failure analysis, such as Product version, FW version, manufacturing date code, .. and etc.

Note:

The *Information Zone* is just the name of the range where the ISP Programmer can read out data even the target is locked or scrambled.

### 5.1 Definition of the Information Zone

The **Information zone** is defined in the 256 bytes just prior to the ISP-memory, as listed below.

For MPC89L(E)51/52/53, it is located at *0x3700~0x37FF*.

For MPC89L(E)54/58/515, it is located at *0xF700~0xF7FF*.

For MPC82L(E)52, it is located at *0x1B00~0x1BFF*.

For MPC82L(E)54, it is located at *0x3700~0x37FF*.

For MPC82G516, it is located at *0xFB00~0xFBFF*.

For MG84FL54, it is located at *0x3B00~0x3BFF*.

For MG84FL516, it is located at *0xFB00~0xFBFF*.

For MG87FL(E)51/52, it is located at *0x1B00~0x1BFF*.

An example of the **Information zone** of MPC89L(E)54/58/515:

|            |                                                   |                  |
|------------|---------------------------------------------------|------------------|
| 0000f640h: | C9 C8 C7 C6 C5 C4 C3 C2 C1 C0 BF BE BD BC BB BA ; |                  |
| 0000f650h: | B9 B8 B7 B6 B5 B4 B3 B2 B1 B0 AF AE AD AC AB AA ; |                  |
| 0000f660h: | A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 9F 9E 9D 9C 9B 9A ; |                  |
| 0000f670h: | 99 98 97 96 95 94 93 92 91 90 8F 8E 8D 8C 8B 8A ; |                  |
| 0000f680h: | 89 88 87 86 85 84 83 82 81 80 7F 7E 7D 7C 7B 7A ; |                  |
| 0000f690h: | 79 78 77 76 75 74 73 72 71 70 6F 6E 6D 6C 6B 6A ; |                  |
| 0000f6a0h: | 69 68 67 66 65 64 63 62 61 60 5F 5E 5D 5C 5B 5A ; |                  |
| 0000f6b0h: | 59 58 57 56 55 54 53 52 51 50 4F 4E 4D 4C 4B 4A ; |                  |
| 0000f6c0h: | 49 48 47 46 45 44 43 42 41 40 3F 3E 3D 3C 3B 3A ; |                  |
| 0000f6d0h: | 39 38 37 36 35 34 33 32 31 30 2F 2E 2D 2C 2B 2A ; |                  |
| 0000f6e0h: | 29 28 27 26 25 24 23 22 21 20 1F 1E 1D 1C 1B 1A ; |                  |
| 0000f6f0h: | 19 18 17 16 15 14 13 12 11 10 0F 0E 0D 0C 0B 0A ; |                  |
| 0000f700h: | 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D ;    |                  |
| 0000f710h: | 49 6E 66 6F 72 6D 61 74 69 6F 6E 3A 20 20 20 20 ; | Information:     |
| 0000f720h: | 31 29 20 50 72 6F 64 75 63 74 20 54 79 70 65 3A ; | 1) Product Type: |
| 0000f730h: | 3D 3E 20 55 52 43 5F 52 35 20 20 20 20 20 20 20 ; | => URC_R5        |
| 0000f740h: | 32 29 20 4D 66 67 2E 20 50 6C 61 6E 74 3A 20 20 ; | 2) Mfg. Plant:   |
| 0000f750h: | 3D 3E 20 47 75 6D 69 2F 4B 6F 72 65 61 20 20 20 ; | => Gumi/Korea    |
| 0000f760h: | 33 29 20 4D 66 67 2E 20 44 61 74 65 3A 20 20 20 ; | 3) Mfg. Date:    |
| 0000f770h: | 3D 3E 20 32 30 30 36 2F 30 31 2F 30 31 20 20 20 ; | => 2006/01/01    |
| 0000f780h: | 34 29 20 46 2F 57 20 56 65 72 73 69 6F 6E 3A 20 ; | 4) F/W Version:  |
| 0000f790h: | 3D 3E 20 31 2E 30 30 20 20 20 20 20 20 20 20 20 ; | => 1.00          |
| 0000f7a0h: | 2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ;    | .                |
| 0000f7b0h: | 2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ;    | .                |
| 0000f7c0h: | 2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ;    | .                |
| 0000f7d0h: | 4D 43 55 20 54 79 70 65 3A 20 20 20 20 20 20 20 ; | MCU Type:        |
| 0000f7e0h: | 4D 65 67 61 77 69 6E 20 20 20 20 20 20 20 20 20 ; | Megawin          |
| 0000f7f0h: | 4D 50 43 38 39 4C 35 34 2F 35 38 2F 35 31 35 20 ; | MPC89L54/58/515  |

User can use any HEX editor to edit his Information Data.

Information Data



Address



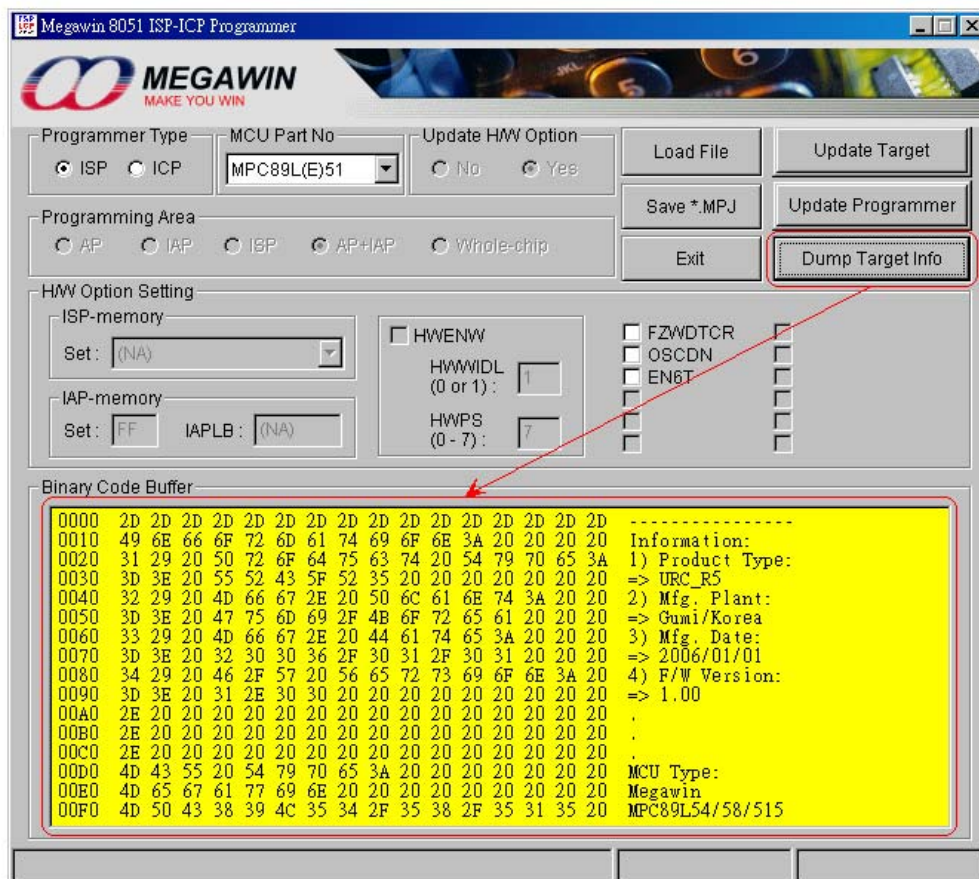
ASCII type



Text type

## 5.2 Dump the Information Data

To view the Information Data in the Information Zone, click **"Dump Target Info"**. Now, the Information Data read by the ISP Programmer will be displayed on the Binary Code Buffer, as shown below.



## 6 Special Notes for ISP

For the ISP operation, the Target MCU's **P3.1** pin is used as the **DTA** pin. The best case is that P3.1 dedicates itself to the ISP operation. However, P3.1 can have its normal function while not in ISP operation as long as the user follows the rule:

**When the ISP Programmer is not connected, the state on P3.1 must be logic-1 when the MCU is just powered on.**

It is because when the MCU is powered on and boots from ISP-memory, the MCU will check P3.1's state to determine which action will be taken: (1) keep running the ISP code, or (2) re-boot to run user's application code. If logic-0 is read, it means ISP operation is requested by the user, and the MCU will take action (1) for further ISP processing. If logic-1 is read, the MCU will take action (2). During ISP processing, P3.1 functions for bi-directional data transfer. It may output logic-1 or logic-0, and also, the ISP Programmer may send it logic-1 or logic-0. So, the user should check if it is harmful to the device/component which is connected to P3.1.

The following figures show the restriction on typical applications of P3.1 if P3.1 also functions for DTA-pin of ISP.

Figure 1:  
P3.1 is used to drive an NPN transistor.

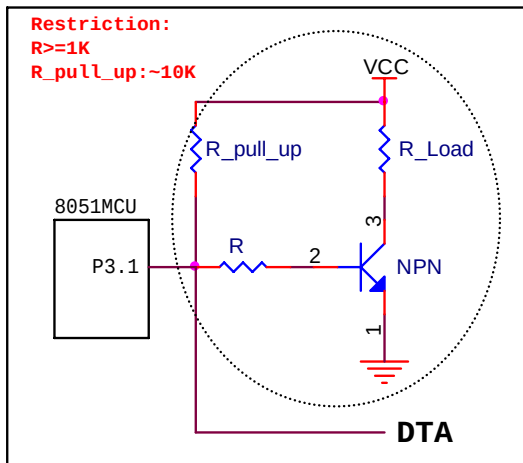


Figure 2:  
P3.1 is used to drive a PNP transistor.

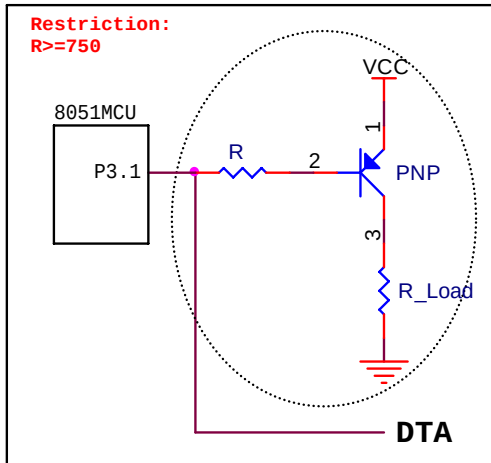


Figure 3:  
P3.1 is used to drive an LED.

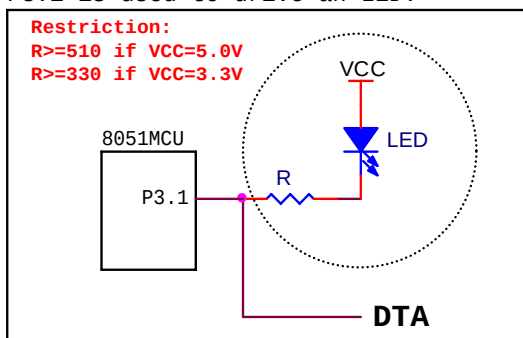


Figure 4:  
P3.1 is used as its original TXD function.

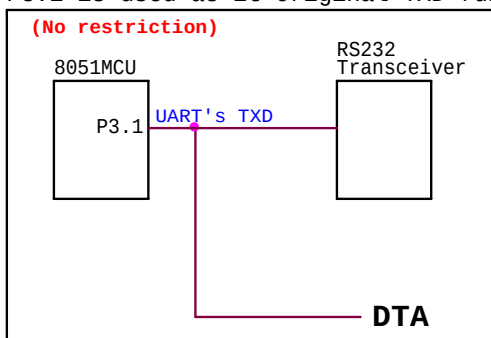


Figure 5:  
P3.1 is pulled low.

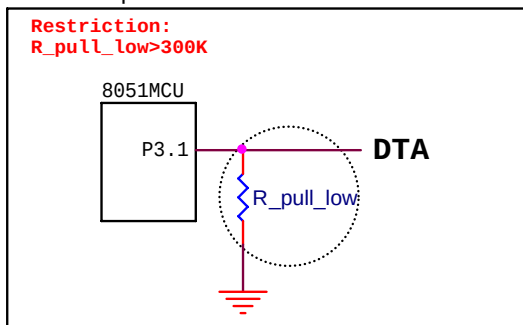
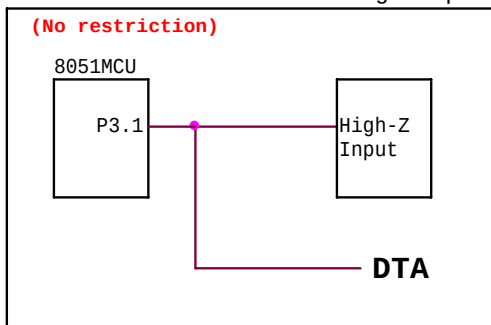


Figure 6:  
P3.1 is used to drive a high-impedance input.

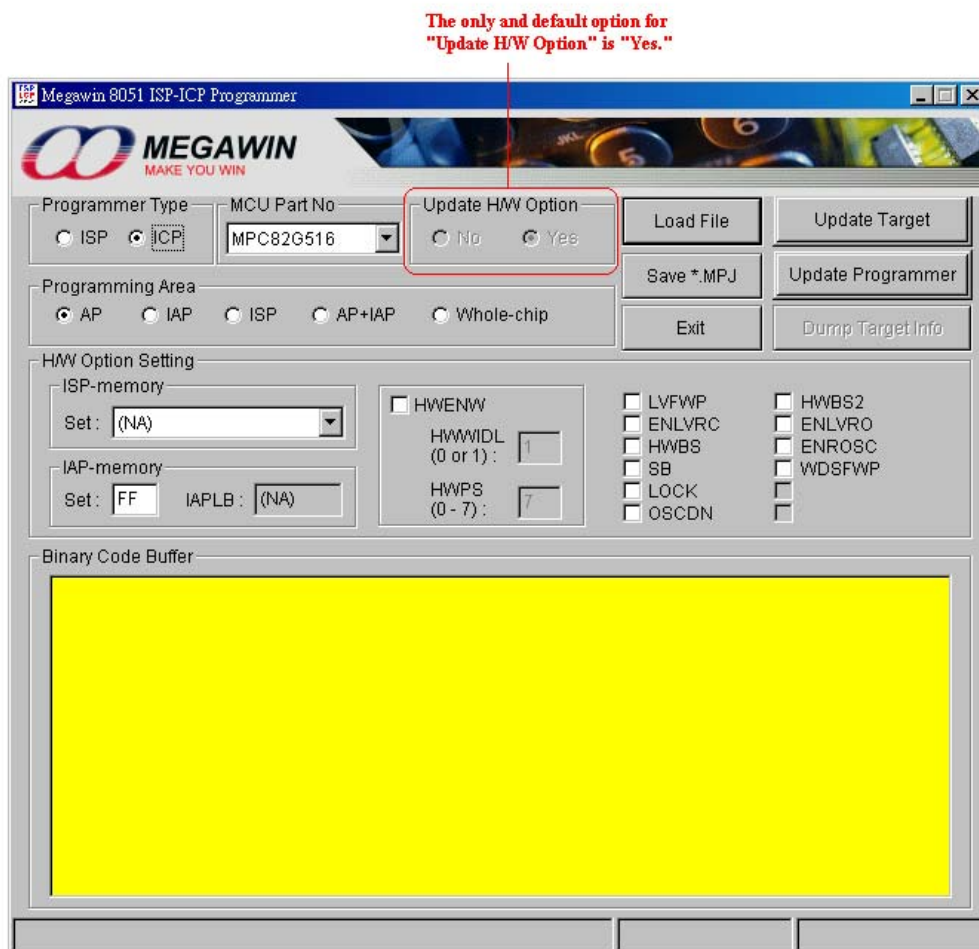


## 7 Special Notes for ICP

When the Programmer is used as an ICP Programmer, the user can select any of the Target Programming Area. This feature, however, is not available if acting as an ISP Programmer. In addition, if IAP-memory or ISP-memory is included in the programming area, the corresponding memory space should be configured by properly setting the H/W option regarding the IAP space and ISP space.

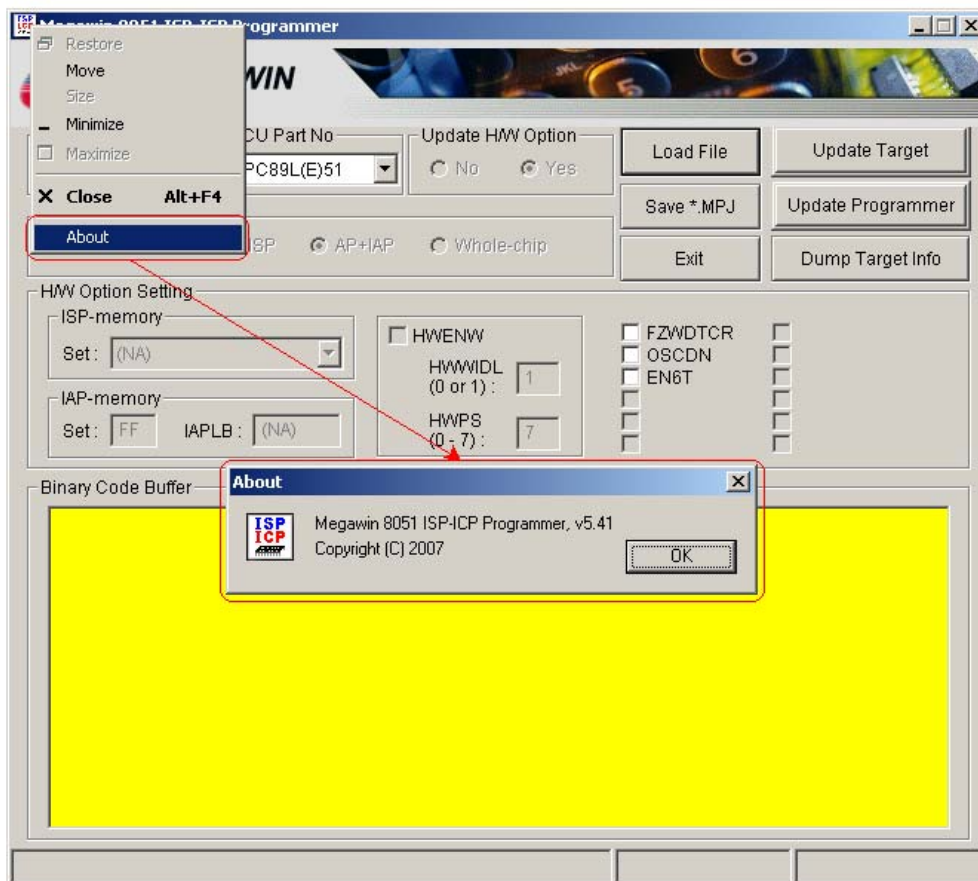
When chip was not locked previously, the user can individually re-program any area he wants. **However, when the chip had been locked previously, any re-programming will be inhibited.** This is for security consideration. It prevents from any re-programming of a code with the malicious purpose. So, re-programming must be inhibited when chip has been locked.

**The only and default option for "Update H/W Option" is "Yes."**



## 8 About Information

Right click on the Title bar of Megawin 8051 ISP-ICP Programmer application, and it will show up the About Dialog.



## Revision History

| Revision | Description                                                                                                                                                                                                                                               | Date       |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| v5.10    | Add auto-upgrade function for the firmware of the Programmer's MCU.                                                                                                                                                                                       | 2007/06/23 |
| v5.20    | (1) Update the included standard ISP-code:<br>"ISP_Code_v5.00.BIN": for MPC89-series & MPC82-series<br>"ISP_Code_v5.00_MG84.BIN": for MG84-series<br>(2) Update the driver to "0E6A030D_8051ISPICP_v3.00.inf" for Vista OS.                               | 2007/07/17 |
| v5.30    | (1) Fix the AP bug: Firmware upgrade will fail if host uses OHCI chip set.<br>(2) For ICP Programmer, the user can select whether to update the hardware option or not.                                                                                   | 2007/10/05 |
| v5.30a   | (1) Fix the AP bug: Wrong display in the H/W Option area when "MPC89E58.MPJ" is loaded.<br>(2) Modify description for HWENW. (Section 3-3-3 & 3-3-4)<br>(3) Correct description for ENLVRO, from 3.8V to 3.7V. (Section 3-3-3)<br>(4) Update Section 2.1. | 2008/01/24 |
| v5.40    | Add support for MG87FL51/52 and MG87FE51/52 in the ISP Programmer.                                                                                                                                                                                        | 2008/07/09 |
| V5.41    | (1) In the Programming Area, the radio button name of "AP+IAP+ISP" changes to "Whole-chip."<br>(2) When operate "Update Target" and "Update Programmer" functions, the only default option of "Update H/W Option" is "Yes."                               | 2008/11/13 |
| V5.42    | Add support for MG87FL(E)2051/4051/6051 in the ISP Programmer.                                                                                                                                                                                            | 2009/01/06 |
| V5.43    | Add support for MG82FG216/232/248 in the ISP Programmer.                                                                                                                                                                                                  | 2009/02/04 |
| V5.44    | (1) Add the "Set Max count " variable in Update Programmer function.<br>(2) Add the "8051 ISP Writer U2" tool description                                                                                                                                 | 2009/02/27 |
| V5.45    | MG82FG216, MG82FG232, and MG82FG248 now have following option bit available for user to configure: ENLVRC, ENLVRO, OSCDN, ENROSC, HWENW                                                                                                                   | 2009/03/16 |
| V5.50    | Add support for MG87FL(E)04 and MG82FL(E)308/316 in the ISP Programmer.                                                                                                                                                                                   | 2009/09/01 |
| V5.51    | 1. Supported the Multi-Load file function.<br>2. Available in Windows 7 operation system.                                                                                                                                                                 | 2010/01/20 |
| V5.60    | Add support for MG82FL(E)532/564 in the ISP Programmer.                                                                                                                                                                                                   | 2010/03/26 |
| V5.61    | Special Release                                                                                                                                                                                                                                           | 2010/06/02 |
| V5.70    | Add support for MG82FE216 in the ISP Programmer.                                                                                                                                                                                                          | 2010/07/05 |
| V5.80    | Add support for MG82FE632/664 in the ISP Programmer.                                                                                                                                                                                                      | 2010/12/22 |
| V5.81    | Modify auto-upgrade function for the firmware of the Programmer's MCU.                                                                                                                                                                                    | 2011/01    |
| V5.82    | The ISP-ICP programmer supports <b>Serial Number</b> function when operate in Mode-3                                                                                                                                                                      | 2011/02    |
| V5.83    | Correct the IAP setting error on ICP programmer                                                                                                                                                                                                           | 2011/04    |
| V5.90    | 1. Add support for MG86FL(E)104 in the ISP Programmer.<br>2. Remove the MG84FL516 in ISP and ICP Programmer.                                                                                                                                              | 2011/05    |
|          |                                                                                                                                                                                                                                                           |            |